The MVC Pattern

- MVC forces separation of concerns
- Domain model and controller logic is *decoupled* from the UI
- HTML is kept apart from the rest of the application
- Maintenance and testing gets simpler and easier

The MVC Pattern

- Interactions with an MVC application follow a natural cycle of user actions and view updates, where the view is assumed to be *stateless*.

- This fits nicely with the HTTP requests and responses that underpin a web application.

The MVC Pattern

- An MVC application will be split into at least three pieces:
- *Models*, which contain or represent the data that users work with. Ideal to aim for: **'Fat Model'**
- *Views,* which are used to render some part of the model as a UI. Ideal: **'Stupid View'**
- *Controllers*, which process incoming requests, perform operations on the model, and select a view to render to the user. Ideal: **'Thin Controller'**

The MVC Pattern

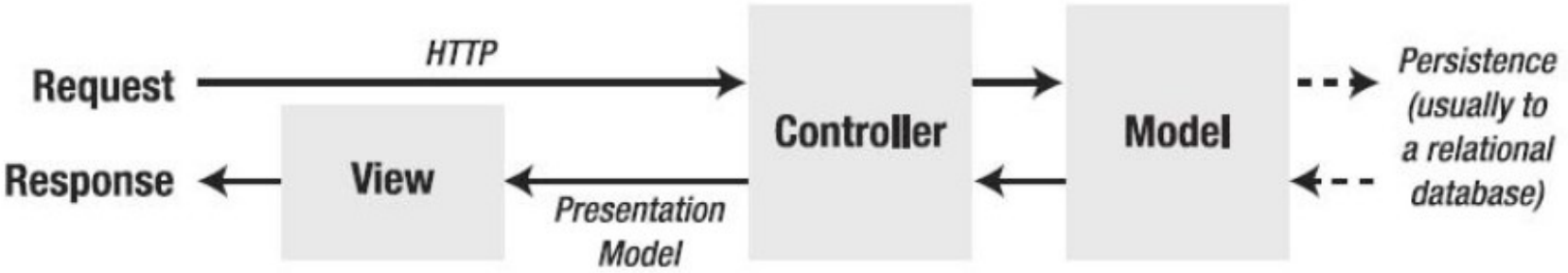

Figure 4-1. *The interactions in an MVC application*

The Smart UI (anti)Pattern

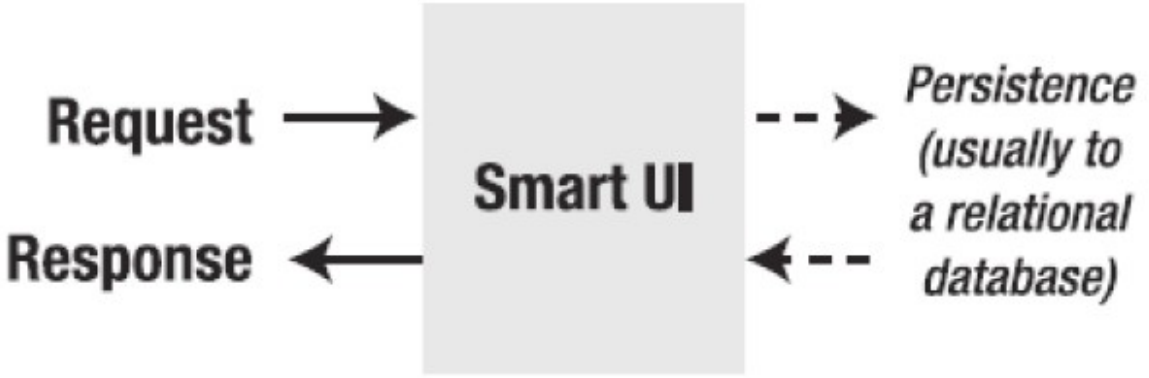

Figure 4-2. The smart UI pattern

The Smart UI (anti)Pattern